

2004 年度修士論文

# 双方向遠隔講義における 演習サポートシステム

提出日： 2005 年 2 月 2 日

指導： 二村良彦教授

早稲田大学大学院 理工学研究科 情報・ネットワーク専攻

学籍番号： 3603U089-0

田中 利幸

# 目 次

第 1 章	序論	5
1.1	研究の背景	5
1.2	研究の目的	6
1.3	本論文の構成	6
第 2 章	既存のシステムとその問題点	7
2.1	既存のシステムとその問題点	7
2.1.1	e-Learning のタイプ別特徴	7
2.1.2	非同期型 e-Learning システムの具体例	8
2.1.3	同期型 e-Learning システムの具体例	9
第 3 章	システムの特徴	11
3.1	システムの概要	11
3.1.1	オーサリングツールの仕様	11
3.1.2	SOBA フレームワーク	14
第 4 章	システムの構成と処理	17
4.1	真偽選択形式教材の双方向化	17
4.2	システムの構成	17
4.2.1	Question Frame(ActionScript)	18
4.2.2	Monitor Script(JavaScript)	19
4.2.3	Relay Applet(JavaApplet)	19
4.2.4	SOBA Synchronizer(Java)	19
4.3	システムの処理	20
4.3.1	イベント送信者側 イベント受信者側の処理の流れ	20
第 5 章	解決すべき問題	22
5.1	解決すべき問題と解決法	22
5.1.1	イベント送信者のイベント発生の重複	22
5.1.2	イベント変更に関する判定の制御	23
5.1.3	教員が 1 人, 生徒が複数人いる時の授業形態	24

第 6 章	遠隔講義モデル	27
6.1	遠隔講義モデル . . . . .	27
第 7 章	結論	29
7.1	まとめ . . . . .	29
7.2	今後の課題 . . . . .	30

# 目 次

2.1	演習問題の画面	8
2.2	解説時の画面	9
3.1	ドラッグ&ドロップ方式	12
3.2	空欄入力方式	12
3.3	ホットオブジェクト方式	13
3.4	複数選択方式	13
3.5	真偽選択方式	14
3.6	SOBA フレームワーク上のアプリケーションの構成	15
4.1	オブジェクトの配置	18
4.2	問題作成中の例	19
4.3	イベント情報の通信 (送信者側 受信者側)	20
5.1	問題演習 (教員 1 人, 生徒 1 人の場合)	25
5.2	問題演習 (教員 1 人, 生徒複数人の場合)	26
6.1	講義風景の一例	27

# 表 目 次

3.1	SOBA+Test Synchronizer と Msn Messenger との比較 . . . . .	16
-----	---	----

# 第1章 序論

現在, 二村研究室では, SOBA フレームワークを利用し, 一方向的な配信を目的とした遠隔講義用の教材を, 教材作成者が特別な変更を加える事なく, 自動的に双方向遠隔講義用の教材に拡張するためのソフトウェア (Contents Controller) の開発に取り組んでいる。

本論文では, Contents Controller の機能の一部として, 遠隔講義の中でも, 問題演習に焦点を定め, 一方向配信用の問題を双方向教材に拡張する問題演習サポートシステム (Test Synchronizer) を構築し, より効果的な問題演習モデルを作成する。

本章では, 本論文の研究背景, 研究目的, 本論文の構成について述べる。

## 1.1 研究の背景

時間と場所の制約を受けない学習方法として e-Learning が注目を集め, 2003 年には 1500 億円規模まで市場が成長したとも言われている。e-Learning の発達に伴い, コンテンツを作成する際のコスト的・技術的面での課題も各種インフラが整備される事により解消され, コンテンツの "量産" が可能となる状況が整ってきた。しかし, この場合の多くのコンテンツは, 生徒への一方向的な配信用のコンテンツであり, より高い効果が期待できる双方向型 e-Learning のコンテンツは, 作成に関して, 一方向型用のコンテンツに比べて, 高いコストや技術を要求される。また, 双方向型 e-Learning においても, その多くは, 10,000 行以上の巨大なソフトウェアであり, 画像データそのものを, 他のピアへ送信する方式の為, 多大なトラフィックがかかってしまい, 実質的には光ファイバーを使ったとしても, 教員 1 人, 生徒 1 人の最低規模である 2 台間での講義を行ったとしても, 快適な環境での講義の展開は保障されない。

そこで, 本研究では, 現在本研究室が京都大学数理解析研究所や, エヌ・ティ・ティ・コム・ウェア株式会社, オムロン株式会社等と共同研究開発を行っている SOBA (ソーバ, Session Oriented Broadband Applications) フレームワークを利用して, e-Learning の中でも, 問題演習に焦点を絞り, 既存の遠隔問題演習システムの不足している部分や問題点を補い, より効果的な遠隔講義モデルを提案する。また, Test Synchronizer を開発することにより, 一方向型 e-Learning 用の教材を教材作成者が変更する事なく, そのまま双方向型 e-Learning にも使用でき, コンテンツ制作のコスト的, 技術的面での課題の解決を目指す。

## 1.2 研究の目的

本研究の目的は、既存の遠隔問題演習システムの問題点を SOBA を利用した演習問題サポートシステムを活用する事で解決し、より効果的な遠隔問題演習システムの作成を目指す。

## 1.3 本論文の構成

本論文は、本章を含む全 7 章からなる。

### 第 1 章 序論

本論文の目的、概要および構成について述べる。

### 第 2 章 従来研究とその問題点

従来研究の代表的な例を挙げ、その問題点について述べる。

### 第 3 章 システムの特徴

従来研究と比較した上で、本システムの特徴について述べる。

### 第 4 章 解決すべき問題

本システムを実現する上で解決すべき問題について述べる。

### 第 5 章 システムの構成と処理

本システムの構成とその処理について述べる。

### 第 6 章 システムの実験結果の評価・考察

本システムの実験結果を提示し、その考察を述べる。

### 第 7 章 結論

本論文のまとめと今後の課題を述べる。

## 第2章 既存のシステムとその問題点

本章では、既存のシステムの代表的な例を挙げ、その問題点について述べる。2.1.1 では、既存の e-Learning のタイプ別特徴を、2.1.2 では、非同期型 e-Learning の事例を、2.1.3 では、同期型 e-Learning の事例について述べる。また、e-Learning の重要な要素である、LMS (ラーニングマネジメントシステム) については、本研究では、触れない範囲なので、本論文の事例においても割愛する。

### 2.1 既存のシステムとその問題点

#### 2.1.1 e-Learning のタイプ別特徴

e-Learning には、非同期型 (非リアルタイム型) と同期型 (リアルタイム型) の 2 タイプに分類され、特徴としては、以下のようになっている。

- 非同期型 (非リアルタイム型)

WWW 教材を活用した個人学習形態のシステム。各自のペースで自由な時間帯に学習できるという利点があるが、教師とのリアルタイムの交流が出来ないことや、強い勉学意欲が要求されるという問題がある。授業をビデオ収録したものをサーバで管理し、受講者が必要に応じてダウンロードして受講するという形態も広く普及している。

- 同期型 (リアルタイム型)

テレビ会議システムを用いた e-Learning において多く行われているディスカッションやディベートに留まらず、ホワイトボードやプレゼンテーションスライドとの連動、アプリケーションの共有、さらにはデータ音声等を同期することで、リアルタイムの双方向コミュニケーションとユーザの分かりやすさを重視している点が特徴である。本研究も、同期型 e-Learning を対象としている。



## 2.1.2 非同期型 e-Learning システムの具体例

### Cultiiva Enterprise Powered by Docent を利用した例

Cultiiva Enterprise Powered by Docent は日本電気株式会社が開発した教材配信による非同期型 e-Learning システム。受講者は、受講したい教材を選ぶ事で、教材をサーバーからダウンロードしてコンテンツを利用する。教材コンテンツの種類としては、以下のものが挙げられる。

- 受講した範囲の教科書にあたる教材資料

図や表等を有効的に使い、理解しやすい構成になっている。また、重要なキーワードは色文字を使うことにより、強調している。

- 教材資料の内容に準じた演習問題

問題形式は、3 個もしくは 4 個の選択肢の中から、正しいもの、もしくは間違っているものを 1 つ選び、その選択肢のラジオボタンを押すことにより、回答する形式（図 2.1）。回答が正解の際には、予め用意された解説の文章が表示される（図 2.2）。誤答の場合には再び、選択肢の中から回答を選ぶことができる。質問したい内容が生じた場合には、管理者へ e-mail にて問い合わせる。

- 確認テスト

問題形式は、演習問題と同様に 3、もしくは 4 択でラジオボタンにチェックする形式。確認テストの場合には、不正解時には、再び他の選択肢を選ぶことは出来ない。正解時にも、解説が表示されない。

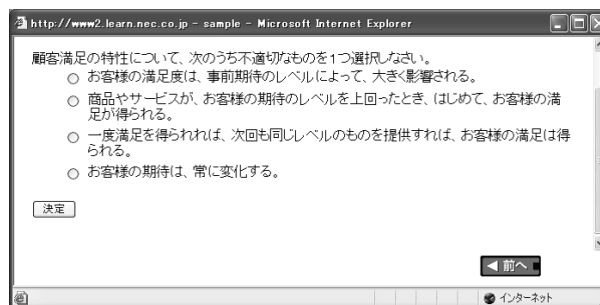


図 2.1: 演習問題の画面

以上の様なコンテンツの特徴から、以下の様な既存の非同期型 e-Learning システムの問題点が考えられる。

1. 個人学習なので、モチベーションが上がりにくい。

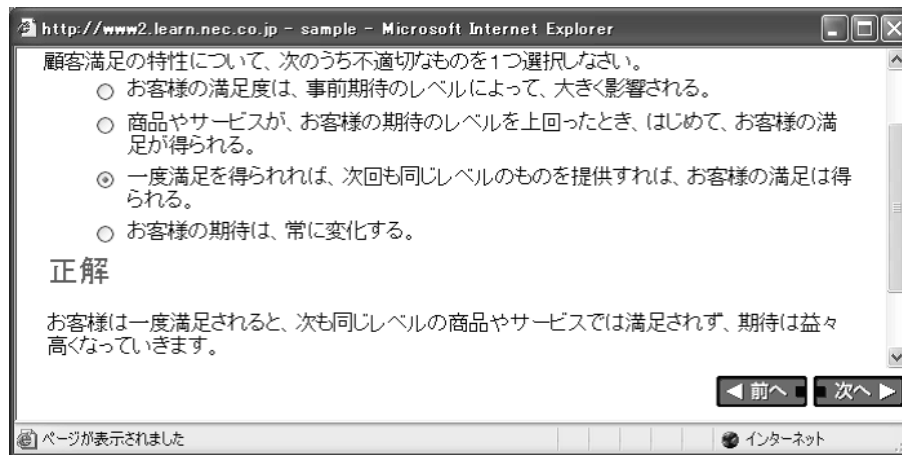


図 2.2: 解説時の画面

2. 教材資料や出題問題, 解説が固定されてしまっている. その為, 受講者の既有能力の差に対応する事が困難である.
3. インタラクティブな支援がないので, 質問が生じた際にリアルタイムでの返答は期待できない.

### 2.1.3 同期型 e-Learning システムの具体例

#### Msn Messenger を利用した例

同期型の e-Learning に Msn Messenger を利用する方式を採用している事例がある。Msn Messenger とは、Microsoft が開発したインスタントメッセージャーの一種であり, その特徴としては以下のものが挙げられる。

- 登録したメンバがオンラインかどうか知る事ができる。
- インスタントメッセージを送ってチャットのように会話できる。
- ファイルの送受信ができる (ただし, 画像データに関しては, データそのものを送受信する方式。
- 複数の人数でインスタントメッセージがやり取りでき, 会話できる。
- NetMeeting やゲーム等のプログラムに招待して一緒にできる。

以上の様な特徴から, 次のような問題点が挙げられる。

1. 画像データがデータそのものを送受信する方式のため、Flash 等の動画ファイルを共有する際には, そのファイルを圧縮して送信する。その為に, 受信側で

は、画像劣化が顕著に発生し、遠隔講義に使用できるクオリティには、程遠い。また、ファイルを圧縮しないで送信することも、設定を変更する事により可能だが、動画では送信するデータ量が大きくなりすぎて、トラフィックの増大により、光ファイバー環境でさえも、2台間の共有においてさえ、顕著なフレーム落ちが発生する。

2. 利用者が、システムを拡張することが困難な為、元々与えられたアプリケーションしか利用する事しかできない。

## 第3章 システムの特徴

本章では、本システムの特徴について述べ、前章のに挙げたシステムとの比較について述べる。

### 3.1 システムの概要

本システムでは、Macromedia Flash MX Professional 2004に搭載されている、クイズテンプレート（オーサリングツール）により、作成された非同期型 e-Learning 用教材について、SOBA フレームワークを利用し、演習問題サポートシステム (Test Synchronizer) を作成することにより、同期型 e-Learning 用教材への自動的な拡張を実現する。

#### 3.1.1 オーサリングツールの仕様

Macromedia Flash MX Professional 2004 に搭載されている、クイズテンプレートでは、教材作成者は、プログラミング等の技術を必要とする事なく、簡単に以下の形式の問題を作成することができる。

1. ドラッグ&ドロップ方式  
質問に対して選択群中のオブジェクトを回答群のターゲットにドラッグ&ドロップすることで、回答するタイプのインタラクション。(図 3.1)
2. 空欄入力方式  
質問に対して、一致する語句をテキストフィールドに入力することで、回答するタイプのインタラクション。(図 3.2)
3. ホットオブジェクト方式  
画面上のオブジェクトをクリックすることで回答するタイプのインタラクション。(図 3.3)
4. 複数選択方式  
質問に対して、複数のチェックボックスを選択することで、回答するタイプのインタラクション。(図 3.4)
5. 真偽選択方式  
質問に対して、真（正しい）または偽（正しくない）のラジオボタンを選択す



図 3.1: ドラッグ＆ドロップ方式

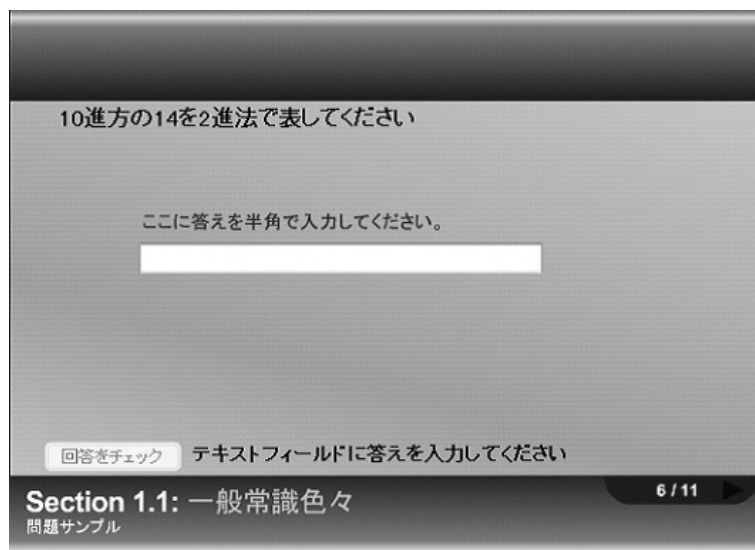


図 3.2: 空欄入力方式

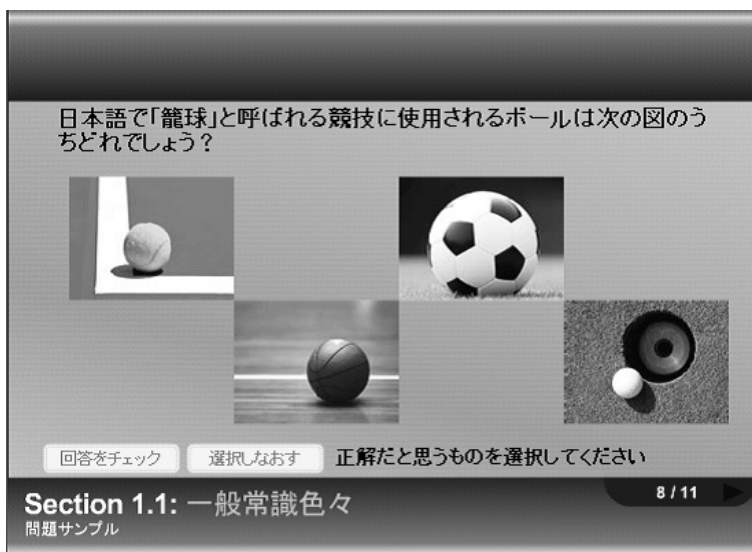


図 3.3: ホットオブジェクト方式

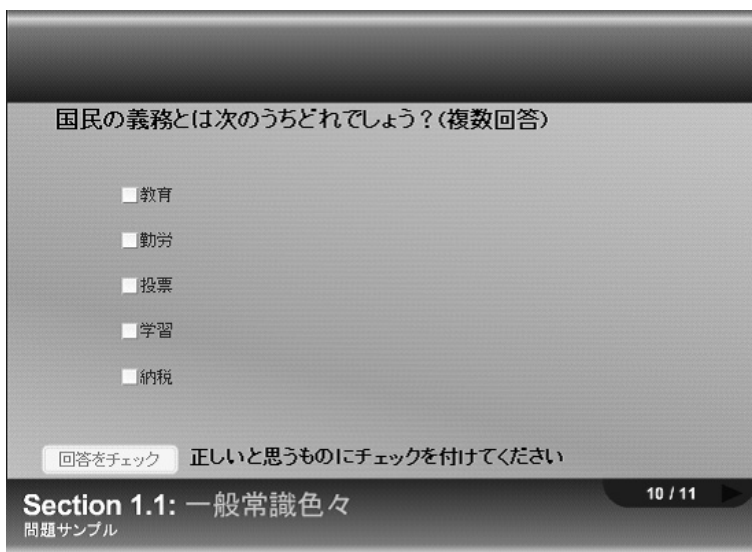


図 3.4: 複数選択方式

ることで、回答するタイプのインタラクション。(図 3.5)

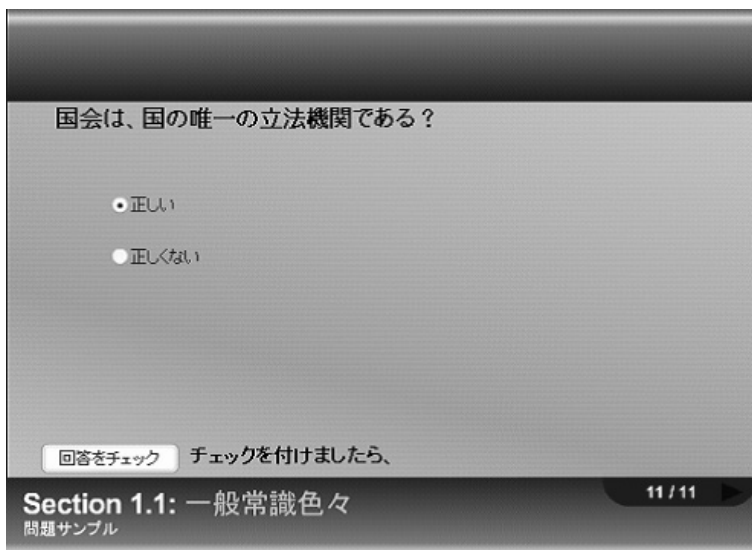


図 3.5: 真偽選択方式

以上の5つの形式について、教材作成者の作成した教材を Test Synchronizer を使用することにより、同期型 e-Learning 用の教材に拡張する。

### 3.1.2 SOBA フレームワーク

本システムでは、SOBA フレームワークを利用し、Test Synchronizer を SOBA の部品アプリケーションとして、構築することで、双方向遠隔講義に必要となるコンポーネントの同期を実現する。以下、SOBA フレームワークについての説明、特徴、2.1.3 で述べた、Msn Messenger との比較について述べる。

#### SOBA とは

SOBA(Session Oriented BroadBand Applications) とは、ブロードバンドを前提としたネットワーク上で P2P で複数のピアを接続し、各ピアがコミュニケーションを行う為の共有空間の構築と運営を可能にする Java ベースのフレームワークである。SOBA フレームワークでは、異なる端末を操作する複数のユーザーに対して、「セッション」と呼ばれる情報を共有する為の空間が提供される。例えば、ビデオカメラ・アプリケーションを共有すると、複数の参加者の間でテレビ電話のようなコミュニケーションが、ホワイトボードアプリケーションを共有すると、絵や文字によるコミュニケーションができる。SOBA フレームワークは、特定の用途のアプリケーションを想定したフレームワークではなく、その上に各種のアプリケーション

が稼動するように、非常に柔軟な設計がなされており、さらにオープンソースとして提供されているので、誰でも機能を付加し、実際に稼動させ改善できる。必要ならば特定のアプリケーションに適するように変更できる。アプリケーション開発者には、ネットワーク通信、リソースの配信、アプリケーションの同期、セキュリティといったネットワークアプリケーション開発に必要な機能を簡単に利用する為の各種APIが提供される。

### アプリケーションの構成

SOBA フレームワーク上に作成できるアプリケーションには、セッションアプリケーションと部品アプリケーションが存在する。セッションアプリケーションは、セッションの初期化や参加条件を規定するなどのセッション自身の振舞いを制御するアプリケーションであり、部品アプリケーションはその配下で同期実行モデルにより全参加に共有されるアプリケーションである。(図 3.6)

本システムでは、SOBA Synchronizer という部品アプリケーションを作成し、Test Synchronizer のモジュールの一部とする。(4.2.4)

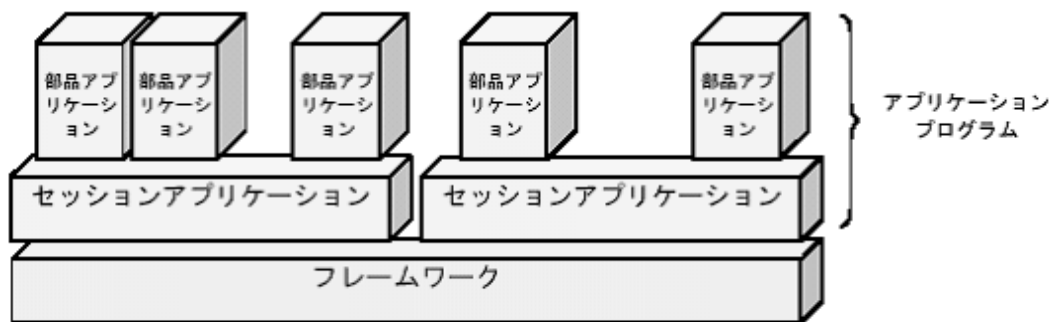


図 1.2 フレームワーク上のアプリケーションの構成

図 3.6: SOBA フレームワーク上のアプリケーションの構成

### SOBA フレームワークの特徴

遠隔問題演習指導を行う視点から見た、SOBA フレームワークの特徴は以下の様なものが挙げられる。

- ビデオカメラや、ホワイトボード等のアプリケーションを教員と生徒が共有することにより、同期型 e-Learning が実現できる。これにより、2.1.2 で挙げた非同期型 e-Learning システムの問題点について解消できる。



	SOBA+Test Synchronizer	Msn Messenger
画面の同期	フレーム落ちは発生せずに同期がとれる.	激しく動くアニメーションの場合, フレーム落ちが生じ, 動作に追いつかない. 完璧な同期がとれない.
教材の音声の同期	途切れる事なく同期がとれる.	同期がとれない.
操作者	教材を共有してる人なら誰でも操作できる	教員と生徒の操作を明示的に切り替えなくてはならない.
教員-生徒の人数	多数-多数	1人-1人
画面のプライバシー	プライバシーを守るような設計ができる.	教員が自分のブックマークリスト等を隠すことができない.

表 3.1: SOBA+Test Synchronizer と Msn Messenger との比較

- Flash や PDF, PowerPoint 等のコンテンツの双方向化を可能とさせるアプリケーションを容易に開発できる。1,000 行程度のプログラムで可能。
- 共有方式が, イベント情報のみを共有する方式なので, ネットワークへの負荷も小さく, 光ファイバー環境なら, 百台以上の遠隔講義も可能である。
- 例えば, 当システム Test Synchronizer と併用することによって, Macromedia Flash MX Professional 2004 に搭載されている, クイズテンプレートで作成した, 非同期型 e-Learning 用教材を, そのまま, 同期型 e-Learning 用教材へ流用できるので, 教材作成者の労力や教材作成コストがかからない。

#### Msn Messenger を用いた遠隔講義との比較

上記をもとに, SOBA フレームワークを利用して作成された, Test Synchronizer と, Msn Messenger で同様の教材を用いて, 遠隔講義を行った場合の比較表を 3.1 に示す。

表に示されている通り, SOBA フレームワークを利用することにより, 既存のシステムの問題点を解消するような, より効果的な演習問題サポートシステムの構築が実現できる。

## 第4章 システムの構成と処理

本章では、本システムの構成とその処理について述べる。本システムでは、Macromedia Flash MX Professional 2004 に搭載されている、クイズテンプレート（オーサリングツール）により、作成された教材を対象としている。本論文では、3.1.1 で紹介したオーサリングツールの5つの問題形式のうち、真偽選択形式を例にとり、本システムの構成とその処理について述べる。

### 4.1 真偽選択形式教材の双方向化

真偽選択形式教材の双方向化を実現する為には、以下の3種類のオブジェクトについて、同期をとらなければならない。

- RadioButton  
生徒が回等する為のコンポーネント。
- ControlButton  
生徒がチェックした RadioButton をもとに正誤を判定し、正誤結果を表示するフィードバックテキストエリアへ正誤結果を渡す。
- NextButton  
次の問題へ移るために次のフレームへ進み、停止させる命令を出す。

以上の3種類のオブジェクトは、基本的に生徒が操作をすることで、教員にも共有されることを想定している。もし、生徒の答えが間違っていたり、生徒が悩んで回答を出すのに時間がかかってしまった際には、教員がリアルタイムに解説を加える。（図 4.1）にオブジェクトの配置図を載せる。

### 4.2 システムの構成

本システム Test Synchronizer は以下の4つのモジュールから構成される。

- Question Frame(ActionScript,64KB)

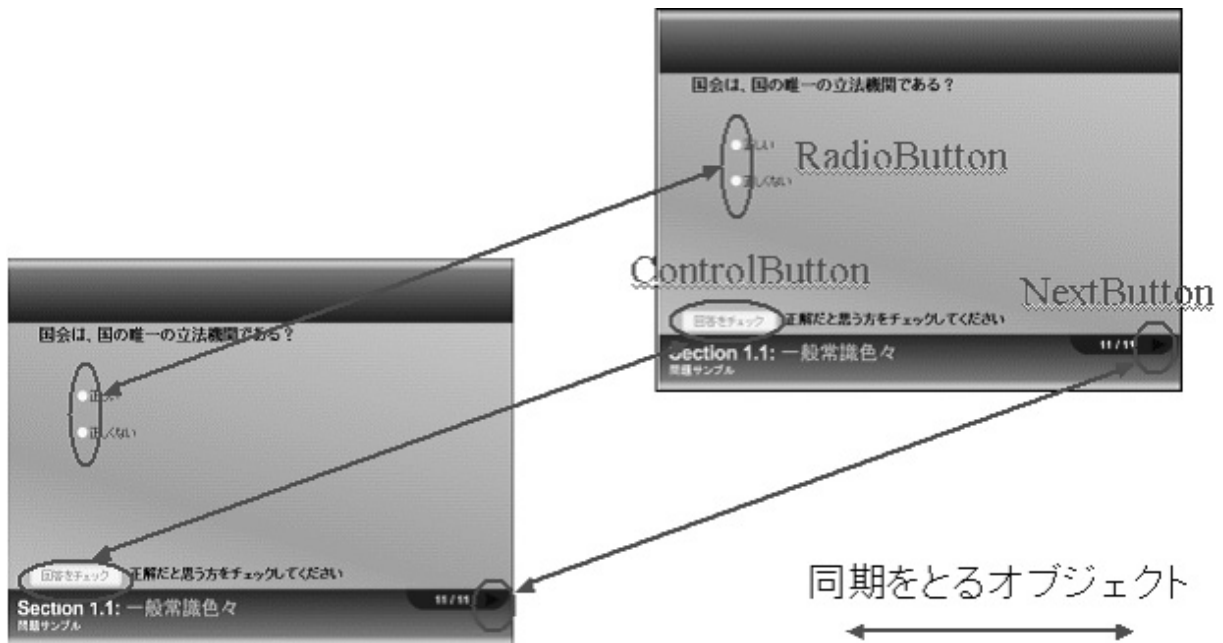


図 4.1: オブジェクトの配置

- Monitor Script(JavaScript,5.74KB)
- Relay Applet(JavaApplet,12.8KB)
- SOBA Synchronizer(Java,22.2KB)

それぞれのモジュールについての説明・機能については以下の通りである。

#### 4.2.1 Question Frame(ActionScript)

- コンテンツ制御ツール
- 問題, 選択肢, 解答等を入力する欄を持ち, 教材作成者はこの欄に入力することで, 簡単に問題を作成できる。(図 4.2)
- 回答用の RadioButton、生徒の回答の正誤を判定する ControlButton、次の問題を表示させる NextButton 等が 用意されており、Flash コンテンツの制御を行う。
- RadioButton,ControlButton,NextButton のイベント情報を Monitor Script に送信する。
- Monitor Script からイベント情報を受け取り、Flash コンテンツにそのイベントを反映させる。

真または偽

インタラクション ID

質問

不正解の選択肢

正解

☒

☐

図 4.2: 問題作成中の例

#### 4.2.2 Monitor Script(JavaScript)

- イベント監視ツール
- Question Frame から送られてきたイベント情報を Relay Applet へ渡す.
- Relay Applet のイベント発生に関するパラメータを監視し, イベントの発生 (パラメータの変更) が認められた際には, その内容を Question Frame に送信する.

#### 4.2.3 Relay Applet(JavaApplet)

- イベント中継ツール
- Monitor Script から送られてきたイベント情報を SOBA Synchronizer に渡す.
- SOBA Synchronizer からイベント情報が送られてきた際には, イベント発生に関するパラメータを変更する.

#### 4.2.4 SOBA Synchronizer(Java)

- イベント共有ツール

- SOBA フレームワークの部品アプリケーションとして作成することで、Relay Applet から送られてきたイベント情報を（全てのピアの間で）共有させる。18個の class ファイルで構成される。
- 同期がとられたイベント情報は、各ピアで Relay Applet に渡す。

## 4.3 システムの処理

### 4.3.1 イベント送信者側 イベント受信者側の処理の流れ

4.2 で挙げた、各モジュールの処理の流れは（図 4.3）のようになる。

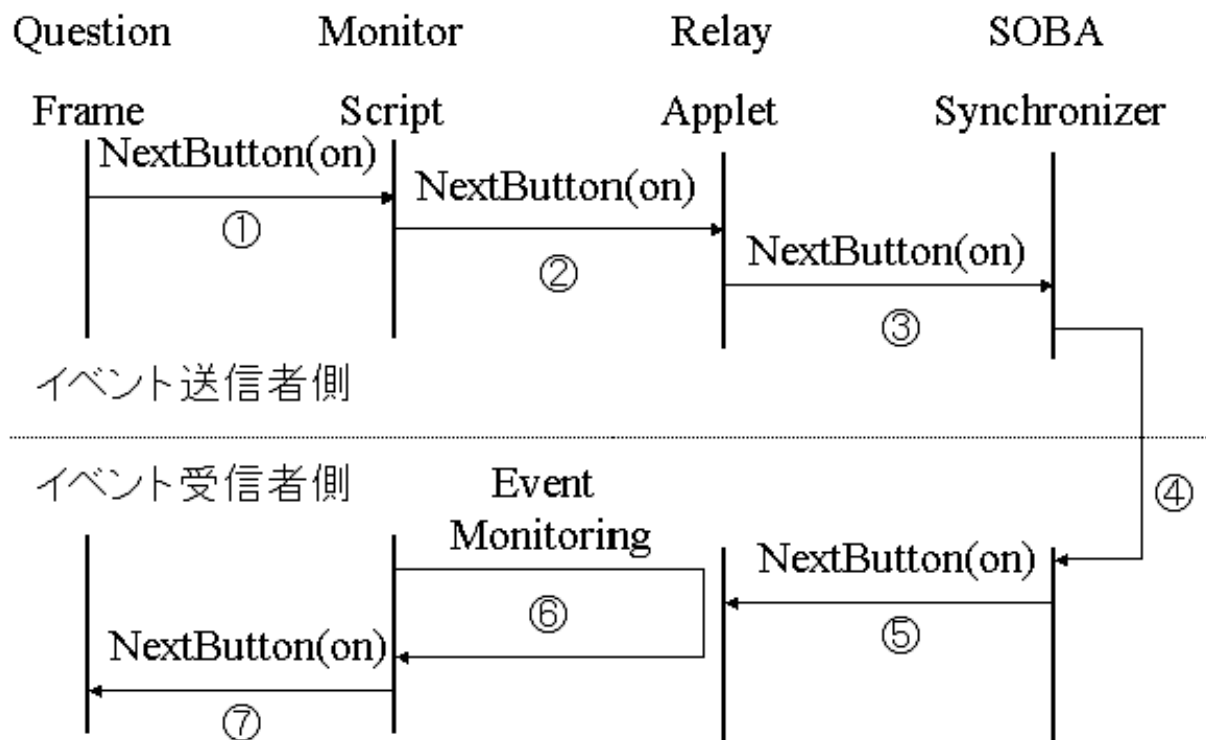


図 4.3: イベント情報の通信 (送信者側 受信者側)

例えば、生徒側が NextButton を押すことにより、次のフレーム（問題）へ移動して、フレームを停止させる処理を行うと、教員も次のフレーム（問題）へ移動して、フレームも停止する。（図 4.3）でいうと、生徒がイベント送信者側、教員がイベント受信者側ということになる。

具体的には、生徒側は

1. Question Frame はイベント情報を含んだ制御コード（今回の例では NextButton(on)）を Monitor Script に送る。…

2. Monitor Script は、制御コードを引数 (今回の例では NextButton(on)) として、Relay Applet のメソッドを呼び出す. …
3. Relay Applet は SOBA Synchronizer に制御コードを送る. 本システムでは、SOBA Synchronizer をサーバー側、Relay Applet をクライアント側と想定し、イベント情報をコマンドパケットと見立てて、Relay Applet から、SOBA Synchronizer へパケットを送信する. …
4. SOBA Synchronizer は、Relay Applet から送られてきたイベント情報を全てのピア (自ピア, 他ピア) の間で共有させる. …

という、処理を行い、イベントを受け取る全てのピア (生徒側, 教員側両方) の処理は、

1. SOBA Synchronizer は、共有されたイベント情報を含む制御コードを Relay Applet に送る. ここでも、SOBA Synchronizer をサーバー側、Relay Applet をクライアント側と想定し、サーバー側から、クライアント側へパケットを送信する。Relay Applet はイベント情報に従って、イベント発生に関するパラメータを書き換える. …
2. Monitor Script は、Relay Applet のイベントに発生に関するパラメータを監視して、イベントの発生 (パラメータの変更) が生じた際には、そのパラメータ (イベント情報を含む制御コード) を取り出す. …
3. Monitor Script は、Question Frame に制御コードを送る. …
4. Question Frame は、受け取った、イベント情報をもとに、Flash コンテンツ (問題を制御するフレーム) にそのイベントを反映させる.

## 第5章 解決すべき問題

本章では、本システムを実現する上で解決すべき問題とその解決法について述べる。

### 5.1 解決すべき問題と解決法

本システムを実現する上で解決すべき問題として、次の3つが挙げられる。

1. イベント送信者のイベント発生の重複
2. イベント変更に関する判定の制御
3. 教員が1人、生徒が複数人いる時の授業形態

#### 5.1.1 イベント送信者のイベント発生の重複

SOBA フレームワークの性質上、共有されるイベント情報はイベント送信者自身のピアにも反映されてしまう。例えば、生徒が次の問題に移ろうと思い、Questoin Frame 内の NextButton を押した際、現時点でのシステムの構成だと、生徒のピアでは、NextButton を押した時に発生するイベントと、SOBA Synchronizer で同期がとられて、再びイベントが Question Frame に返ってきた際の処理で、2度イベントが発生してしまい、教員は1フレーム先にしか進まない（次の問題へ移る）間に、生徒は2フレーム先に進んでしまう（2問先の問題までとんでしまう）という、問題が生じてしまう。この問題を解決する為に、イベント発生者側では、イベント発生時の処理を取り消して、Monitor Script へ制御コードを送るという動作のみを行うように、Question Script を変更する必要がある。以下に NextButton イベント発生時の処理を以下に示す。

NextButton イベント発生時の処理 (ActionScript)

```
on (release) {  
  
    if (nextButton != "off"){ // 1度目に NextButton 押す時の為の処理  
        nextButton = "on";  
    }  
}
```

```

getUrl("javascript:setNextButton('" + nextButton + "');"); //Monitor Script
への送信部分
}

```

### 5.1.2 イベント変更に関する判定の制御

また、本システムでは、Questoin Frame 内の NextButton を押した際に、nextButton という、変数にイベント情報”on”という文字列を格納し、Monitor Script に渡すことにより、イベント情報を渡す。その”on”というイベント情報が SOBA Synchronizer により、共有空間上の全ピアに反映された時に、Relay Applet 内で、new nextButton という変数に、”off”という文字列を格納することにより、イベントの変更を表す。つまり、Question Frame 側では、変数 nextButton と new nextButton の値を比較し、値が違っていた際に、各フレームイベントを反映させる。この時に、nextButton に”off”を格納することで、イベントの反映を 1 回のみの実行に制御する。

しかし、次にイベントが発生した時には、nextButton に”off”が格納された状態でイベント情報を渡してしまうので、new nextButton は”on”が返って来なければ、Question Frame 側でイベントの反映の命令が実行されない。この問題を解決する為、Relay Applet 内で nextButton が”on”の時は new nextButton が”off”、nextButton が”off”の時は new nextButton は”on”となるように制御を加える必要がある。それに伴い、Question Frame 内でも、new nextButton が”off”の時に nextButton が”off”、new nextButton が”on”の時に nextButton が”on”となるように制御を加える必要になる。

NextButton を押した際に発生するイベント情報の処理に関する、Question Frame (イベント変更受け取り後) と Relay Applet(SOBA Synchronizer から nextButton を渡された際) 内の処理を以下に示す。

#### Question Frame 内での処理 (イベント変更受け取り後,Action Script)

```

if (new_nextButton != nextButton){

if(new_nextButton == "on"){
nextButton = "on";
_root.QuizTrack.setNewPage(); //次フレームを
_root.nav.updateFrame(); //呼び出して,
gotoAndStop(2); //初期化する処理
}
}

```



```

        if(new_nextButton == "off"){
nextButton = "off";
_root.QuizTrack.setNewPage();
_root.nav.updateFrame();
gotoAndStop(2);
        }
}

```

Relay Applet 内での処理 (SOBA Synchronizer から nextButton を渡された際,Java Applet)

```

/**
 * nextButton の値を確認して new_nextButton の値を決定する
 * Relay Applet 内での処理.
 * @param packet コマンドパケット
 * @param commandName コマンド名
 */
if(commandName.equals("NEXT")) {
NextCommandPacket npacket = (NextCommandPacket)packet;
        String nextButton = npacket.getNextButton();
        //パケット内の nextButton の値を格納
timesClicked++;
if(nextButton.equals("on")){
        new_nextButton = "off";
        showDebug(); //new_nextButton に何が格納されたか表示する関数
}

        else
        if(nextButton.equals("off")){
new_nextButton = "on";
showDebug();
        }
}

```

### 5.1.3 教員が1人,生徒が複数人いる時の授業形態

本システムでは, イベント情報を共有するのみなので, 光ファイバー環境においても, 100台以上のマシン間で, 問題を共有することも可能である. しかし, 生徒が複数人いる場合, SOBA フレームワークの性質上, 全員が同一の共有空間を持つ

と、生徒 A が出した回答が、生徒 B にも反映されてしまう。これでは、生徒 B の学習効率は期待できない。そこで、生徒 A と生徒 B は、別の共有空間を持つようにする。この方法で、生徒同士のカンニングも防止でき、効率的な問題演習授業が期待できる。つまり、生徒が  $n$  人参加する授業では、 $n$  個の共有空間を作り、授業を展開する。以上の事を図 5.1、図 5.2 に示す。

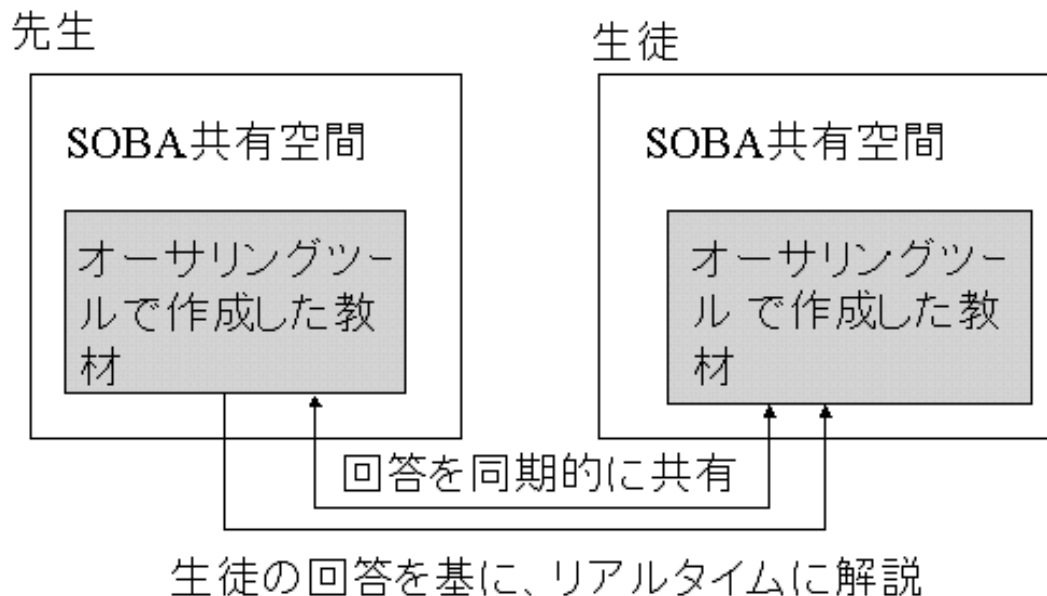


図 5.1: 問題演習 (教員 1 人, 生徒 1 人の場合)

先生1人、生徒複数の場合→生徒数分の共有空間を持つ。

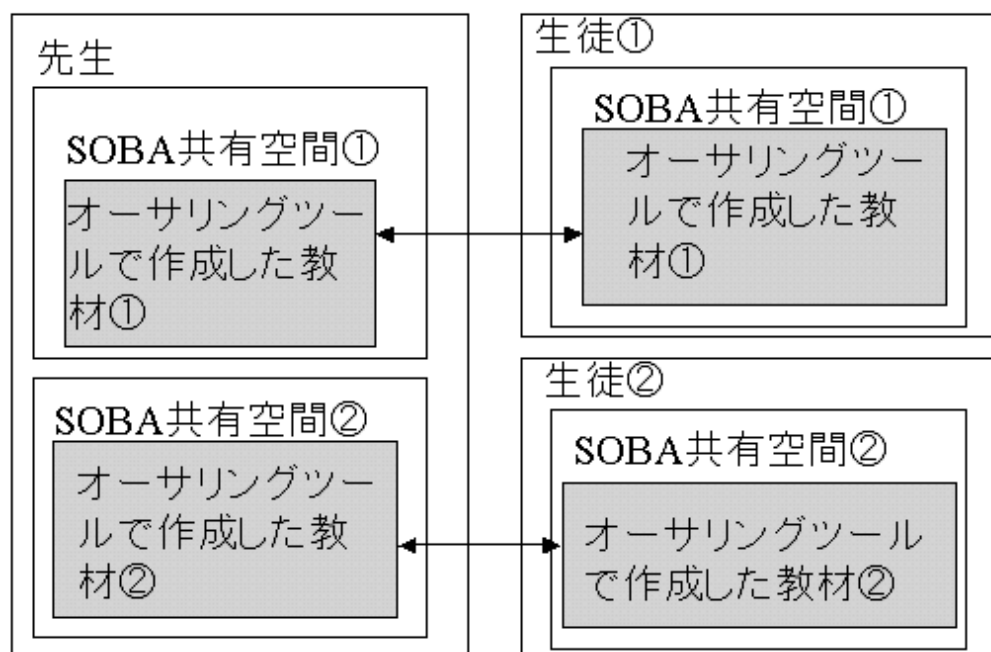


図 5.2: 問題演習 (教員 1 人, 生徒複数人の場合)

## 第6章 遠隔講義モデル

本章では、本研究で実装したシステムを用いて、実際に遠隔講義を行った時を想定して提案する遠隔講義モデルを示す。

### 6.1 遠隔講義モデル

図 6.1 は提案する遠隔講義風景のモデルである。



図 6.1: 講義風景の一例

この図では、教員 1 人、生徒 1 人での授業を想定している。一番外側のウィンドウがセッションウィンドウである。その中で、ビデオチャット、ホワイトボード、Test Synchronizer の機能で作成した教材が共有されている。この図 6.1 を生徒側のウィンドウに表示されているとしたら、教員側のウィンドウにも、同様の 3 つのアプリケーションが表示されていて、同期がとられている。

この状態で、生徒が教材に対してラジオボタンをチェックして、回答をすると、その回答が教員側にもリアルタイムに反映される。ここで、生徒の回答が間違ってい

たり、生徒から質問が生じた際には、その内容についてリアルタイムに教員がホワイトボード等を使って説明を加える。すでに用意された解答や解説を見ることで学習する非同期型 e-Learning システムに比べて、生徒のレベルや質問内容を考慮して、柔軟な返答や解説をすることができるので、より効率的な問題演習指導が期待できる。

## 第7章 結論

本章では、本論文のまとめと本論文で提案したシステムの今後の課題を述べる。

### 7.1 まとめ

既存の e-Learning システムについて 2.1.2, 2.1.3 において、次の様な問題点をあげた。

既存の非同期型 e-Learning システムの問題点

- 個人学習なので、モチベーションがあがりにくい。
- 教材資料や出題問題、解説が固定されてしまっている。その為、受講者の既有能力の差に対応する事が困難である。
- インタラクティブな支援がないので、質問が生じた際にリアルタイムでの返答は期待できない。

既存の同期型 e-Learning システムの問題点

- 画像データがデータそのものを送受信する方式のため、Flash 等の動画ファイルを共有する際には、そのファイルを圧縮して送信する。その為に、受信側では、画像劣化が顕著に発生し、遠隔講義に使用できるクオリティには、程遠い。また、ファイルを圧縮しないで送信することも、設定を変更する事により可能だが、動画では送信するデータ量が大きくなりすぎて、トラフィックの増大により、光ファイバー環境でさえも、2 台間の共有においてさえ、顕著なフレーム落ちが発生する。
- 利用者が、システムを拡張することが困難な為、元々与えられたアプリケーションしか利用する事しかできない。

本論文では、SOBA フレームワークを利用したアプリケーション Test Synchronizer を構築することにより、非同期型 e-Learning 用教材を同期型 e-Learning 用教材へ拡張し、SOBA 共有空間上で動作させる事により、上記の問題点を解消するよ

うな問題演習サポートシステムを作成した。本論文は、3.1.1 で挙げた、オーサリングツールの 5 種類の形式のうち、真偽選択形式の問題の双方向化を実現した。

第 2 章では、既存の e-Learning を 2 種類のタイプに分け、それぞれタイプの具体例を挙げる事によって、その問題点や不足している点を抽出した。

第 3 章では、本システムの特徴を挙げ、既存のシステムと比較することにより、本システムの e-Learning への有効性を示した。

第 4 章では、本システムの構成とその処理の流れについて述べた。

第 5 章では、本システムを実現する上で解決すべき問題とその解決法について述べた。

第 6 章では、本システムを使用した遠隔講義モデルを提案し、本システムの有用性を述べた。

## 7.2 今後の課題

本論文で提案したシステムにおいて、今後解決すべき課題として以下の点が挙げられる。

### 1. 本システムの有効性の実証実験

今回、実装した Test Synchronizer を用いて、実際の遠隔講義実験を行ってみたいことにより、Test Synchronizer の有効性を検証する。また、その時にバグ等が発見された際は随時デバッグを行う。

### 2. オーサリングツールの他形式への対応

限システムでは、オーサリングツールの仕様の中の真偽選択形式しか、対応していない。4 つの形式（ドロップ&ドラッグ形式、空欄入力形式、ホットオブジェクト形式、複数選択形式）で作成された問題の演習も共有出来るように、Test Synchronizer を拡張する。

### 3. ContentsController の実現

ContentsController は、Flash に限らず、PDF や PowerPoint 等、様々な形式の既存の教材の自動的な双方向化を実現するシステムである。現システムでは、Macromedia Flash MX Professional 2004 に搭載されている、クイズテンプレートで作成された教材についての双方向化を実現するが、それ以外の形式の教材についても双方向化できるような、ContentsController の実現を目指す。

今後は以上の課題に取り組み、より有用なシステムの開発を行なう。

# 謝辞

本研究を行なう機会を与えて頂き，また，研究の全過程を通して日頃より暖かい御指導を賜った 二村 良彦 教授, 小西 善二郎 客員講師に心より感謝の意を表します．

そして，公私に渡る様々な相談相手となって頂いた，二村研究室のみなさんに厚く御礼を申し上げます．



## 参考文献

- [1] SOBA PROJET,  
<http://www.soba-project.org/jp/index.html>
- [2] SOBA プロジェクト, ”フレームワーク 1.1 プログラミングガイド”, 2004
- [3] 経済産業省商務情報政策局情報処理振興課, ”e ラーニング白書 2004/2005 年版” オーム社, 2004
- [4] 中島一彰, 川本亜紀子, 大芝崇, 吉坂主旬, 田淵仁浩, NEC インターネットシステム研究所, ”リアルタイム Web 共有方式による双方向コミュニケーション基盤” 情報処理学会 研究報告, 2004 年 1 月 vol.2004 No.002, pp.57-64
- [5] 阿久津智則, 仲田和宏, 山田国裕, 中村俊一郎, 宮地功, 吉田幸二, ”Flash アニメーションを用いたコンピュータリテラシ授業の問題理解支援の試み”, 電子情報通信学会 信学技報 ET2004-41 (2004-10) pp.27-32
- [6] 藤田紀勝, 新井吾朗, 山崎敏載, ”公共職業訓練のための e-learning 教材作成支援システム” 電子情報通信学会 信学技報 ET2004-5 (2004-05) pp.25-30
- [7] 大塚 勝三, ”FLASH ACTIONSCRIPT HANDBOOK 3RD EDITION”, ソフトバンクパブリッシング株式会社, 2004.
- [8] 野中 文雄, ”オブジェクト指向で考える ActionScript”, セレンディップ, 2003
- [9] 宮坂雅輝, ”JAVASCRIPT HANDBOOK 4TH EDITION”, ソフトバンク パブリッシング株式会社, 2003
- [10] RADKRAFT:Java Applet + Signature  
<http://kamakura.cool.ne.jp/oppama/jaco/java-plugin-sig.html>